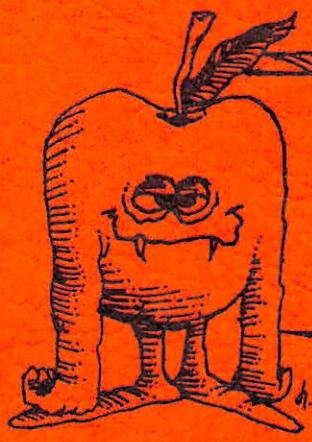


NORTHERN ILLINOIS APPLE USERS GROUP

VOL. 3-NO. 4

THE HARVEST

DECEMBER 1981

MEETING NOTICE

DECEMBER- - - - - - - - - - -DECEMBER 12
JANUARY- - - - - - - - - - - --JANUARY 9

IN THIS ISSUE

CLUB OFFICERS, MEMBERS AIDE, TREASURER'S REPORT- - - - - - - - - - - - - - - - 2
REAL ESTATE ANALYZER, Barry Bayer -3
PRINTING LABELS ECONOMICALLY, E.J Neiburger - - - - - - - - - - - - - - - - - -4
UCLASSIFIED ADS -5
1200 BAUD MODEMS, DON'T GET TOO EXCITED, Terry Cronin - - - - - - - - - - - - -5
RS-232 IS STANDARD, ISN'T IT, Al Guthrie - - - - - - - - - - - - - - - - - - - 6
PERUSING THE PERIODICALS, Marlys Newcome - - - - - - - - - - - - - - - - - - - 8
ANOTHER MARQUEE, Rik Anderson - 9
DECEMBER PUZZLE -11
INPUT,GET, COMMAND? Robert Paarlberg- 12
SIG NEWS- -15
DECEMBER CALENDAR -18
PROGRAM CONTEST ENTRY FORM- -19

CLUB OFFICERS

President-
Dave Alpert---------- 312-295-6078
Vice Pres-
Guy Lyle---------- 312-438-7941
Treasurer-
Jim Wilson---------- 312-358-2196
Secretary-
Joel Kurasch---------- 312-677-8358
Librarian-
Jim Pfeiffer---------- 312-289-7311
SIG Co-Ordinator-
Eric Stral---------- 312-885-1941
Software Chairman-
Rob Stewart---------- 312-537-0544
Hardware Chairman-
Paul Stadfeld--------- 312-359-2378
Program Chairman-
Guy Lyle---------- 312-438-7941

Steeering Committee-
Ted & Mary Rosemann--- 312-338-4833

Club Adresses:

MEMBERSHIP, etc.--------BOX 787,
Palatine, Il, 60067

NEWSLETTER----------Terry Tufts,
1015 S.Ridge Ave,
Arlington Heights, Il,
60005.
312-392-7735

BULLETIN BOARD----------312-295-6926 daily as
available and after
10:30pm local time.

Membership is open to all. Dues are $12.00 annually with a one time initiation fee of $5.00 at the time of admission. An additional fee will be charged to cover the cost of mailing to foreign members. Membership applications are available from the club Secretary at the meetings or by mail.

NIAUG NEWSLETTER is published by the Northern Illinois Apple Users Group and edited by Terry Tufts. Contents may be reproduced by other Apple Clubs for their newsletters or bulletins except where specifically reserved by the author. Permission to copy these articles may be obtained by writing the author through our editorial offices. Please indicate Northern Illinois Apple Users Group, HARVEST, is the source of any material reproduced. Newsletter exchange is welcomed. The HARVEST recognizes that APPLE and APPLE manufactured products are trademarks of the APPLE Computer Company.

MEMBERS AIDE

The members listed below have volunteered to answer questions from club members who need a 'HOT LINE' type answer that can be answered over the telephone. Please try to be brief when you call as a courtesy to them. Their names and phone numbers are listed below. The numbers after their names represent their special talents. PLEASE-NO CALLS at dinner time or after 10 pm.

Paul Stadfeld--------312-359-2378		4-9
Guy Lyle----------312-438-7941		4-9
Earl Allen----------312-837-9259		1-9
Ted Rosemann		
Mary Rosemann--------312-338-4833		2,3
Joel Kurasch--------312-677-8358		9,0
Leon Alexander-------312-725-5309		8,9
Gordon Banks---------312-324-6194		0,8
Bob Noll----------312-888-0808		1,9
Ken Nestle----------312-620-7745		2,3
Herb Schulz----------312-968-6927		0,
Rich Lundeen---------312-420-8486		2,3

Mach. Lang=1	Int. BASIC=2
Applesoft =3	Hardware =4
Arrays =5	DOS =6

All of above=9
PASCAL=0 ,Z80 CARD=8(are not included in the 9 designation)

When numbers appear before a 9, the person is especially qualified in that area as well as in all other areas.

TREASURER'S REPORT

Oct. Ending Bal.	$3102.97
Membership	
26 new members	661.95
19 renew	
Newsletter	<366.18>
Disk Project	
216 sold	
600 purchased	<536.00>
Miscellaneous	
Copy of DB Master for club	
Post Office box rental	
Hall rent	
Stationery & other	<250.35>
November ending Balance	$2612.39

Respectfully submitted,
Jim Wilson, Treasurer,
Nov 14, 1981

REAL ESTATE ANALYZER

Reviewed By Barry D. Bayer

Real Estate Analyzer, published by Howard Software Services, comes on a DOS 3.2 Disk, muffinable to 3.3, a data storage disk, and approximately 40 pages of manual all packaged together in a leatherette binder. List price is $150.

The purported purpose of the program is to enable the user to analyze a projected real estate purchase. Purchase price, loan information, expenses, income, and various projection factors are entered by the user. The program then projects pre-tax and post-tax cashflow for up to 10 years. Finally, the sale of the property is assumed at the end of each of the 10 years and several different "Return on Investment" figures are calculated. The results can be shown on the screen, printed, or both. Figures 1 represents the inputted items required for the analysis. Figure 2, the cash flow analysis, and figure 3, the Return on Investment tables.

A study of these printouts shows a nice careful job of report formatting. The reports really look that good, even on my old IBM selectric printer. But study of the report also reveals some of the program's weaknesses. Real Estate Analyzer is clearly a child of the the late 70's, and 80's, and very clearly comes from the West Coast. This is evident not only from the table of expense items (gardener, pool service), but also from the lack of any direct indication of either cash flow or profit related to cash investment in any given year.

(Figures such as "Cash on Cash" are very important to those who must justify real estate investments in some manner other thatn assuming inflation is going to bail everyone out a few years from now. Note the "After Tax Cash Flow" lie on the bottom of Figure 2. The author presumably considers the deal a reasonable one (note the I-R-R's in Figure 3) but it is still an after cash, let alone before case, loser well into the 10th year, and presumably for another 10. Only the assumed 10 per cent per year inflation in property values---see Figure 1---makes the deal at all attractive.

Another problem with the program is that only 1 rate of projected increase in each analysis factor (income, value, expenses, taxes, etc) is permitted. Thus it is impossible to accurately portray a situation where some tenants can by boosted immediately, some will get rent increases a year from now, but the anchor tenant has lease options to 1990. Similarly, how do you account for sophisticated lease arrangements where rents increase at regular intervals and there are also CPI adjustments? A lot of different rental units can be entered, but the program really is not suited to any but gross analysis of a large, sophisticated purchase.

The program is written in unprotected Applesoft. And there is even some internal documentation in the form of REM statements. (Probably where the author forgot to take them out.) It is not, however, a classic example of structured programming. It took a couple of hours to find the spot for a patch which would permit me to load a second sheet of paper. Into my single feed Selectric (rather than the continuous feed paper the program demands.) But it is possible to do things such as add Cash on Cash percentages, delete the "Pool Expense" line, and the like. A protected, unlistable program would be of much less utility.

With all of these problems, the program works reasonably quickly and well. Entering names and figures is relatively easy, although you have to go through the entire data entry routine just to correct one mistake. It is very hard to crash. The printout is formatted well, is easy to read, and looks rather nice. Howardsoft has even provided a very professional looking cover page printout for the Investment Analysis Report. But, the program doesn't really do much more than you could do with VisiCalc. (Of course, that could be said about a lot of programs, these days.) The manual collects some useful information, and does a decent job of explaining the various investment return calculations. It also refers the reader to some textbooks on the subject. I believe the program to be rather expensive at $150, but if you're thinking of investing in 6 flats, or single condominium units, and haven't figured out how to use VisiCalc, Real Estate Analyzer may be just the ticket.

PROPERTY SPECIFIC INFORMATION:

GENERAL INFO-	
DESCRIPTION OF PROPERTY	SAMPLE RENTAL
DISK-FILING NAME	DEMO
DATE OF ANALYSIS	JANUARY, 1981
PURCHASE INFO-	
PURCHASE PRICE	100000
ESCROW & BANK CHARGES (%)	.5
YEARS SINCE PURCHASE	0
PAST LOAN REDUCTION	0
PAST CASH FLOW	0
CURRENT MARKET VALUE	100000
LOAN INFORMATION-	
NUMBER OF LOANS	1
REMAINING BALANCE OF LOAN #1	80000
INTEREST RATE OF LOAN (%) #1	12
REMAINING YEARS ON LOAN #1	25
METHOD:1=AM:2=INT ONLY'3=OTHER) #1	1
#TOTAL LOANS OUTSTANDING	80000
DEPRECIATION INFO-	
NO. OF DEPR'N ITEMS	1
TOTAL DEPRECIABLE LIFE #1	25
REMAINING DEPRECIABLE LIFE #1	25
TOTAL DEPRECIABLE AMOUNT #1	75000
REMAINING DEPRECIABLE AMOUNT #1	75000
METHOD'1=SL/2=SD/3=DB/4=OTHER) #1	1
RENTAL INCOME INFO-	
NUMBER OF UNITS	1
SCHEDULED MONTHLY INCOME #1	500
VACANCY/COLLECTION LOSS (%)	3
#TOTAL OPERATING INCOME	5820
ANNUAL EXPENSE INFO AS % OF PRICE-	
TAXES	1% OR $1000
INSURANCE	.25% OR $250
UTILITIES	.7% OR $700
ADVERTISING	.01% OR $10
SUPPLIES	.01% OR $10
INTERIOR REPAIRS	.1% OR $100
EXTERIOR REPAIRS	.9% OR $900
POOL SERVICE	0% OR $0
GARDENER	0% OR $0
RUBBISH	0% OR $0
WORKMEN'S COMP	0% OR $0
REPLACEMENTS	0% OR $0
LICENSES	0% OR $0
MANAGEMENT COSTS	0% OR $0
OTHER	0% OR $0
#TOTAL OP'G EXPENSE	2.97% OR $2970

>>>>> ASSUMPTIONS FOR ANALYSIS:

ANTICIPATED % ANNUAL INFLATION IN-	
PROPERTY VALUE	10
OPERATING EXPENSES	7
RENTAL INCOME	5
VALUE OF MONEY (AFTER TAX)	5
PROPERTY TAXES	1
TAX INFORMATION-	
ORDINARY TAX RATE (%)	54.61
CAPITAL GAINS TAX RATE (%)	23.25
SALE COSTS AS % OF PRICE-	
BROKER'S COMMISSION	6
ESCROW FEES	1
OTHER CLOSING COSTS	2

>>>>> CASH FLOW ANALYSIS:

ITEM	%	YEAR 01	YEAR 02	YEAR 03	YEAR 04	YEAR 05	YEAR 06	YEAR 07	YEAR 08	YEAR 09
GROSS INCOME-										
ANNUAL RENTS	5.0	5820	6111	6417	6737	7074	7428	7799	8189	8599
OPERATING EXPENSES-										
TAXES	1.0	1000	1010	1020	1030	1041	1051	1062	1072	1083
OTHER EXPENSES	7.0	1970	2108	2255	2413	2582	2764	2956	3163	3385
#TOTAL OP'G EXPENSES	3.0	2970	3118	3276	3444	3623	3814	4018	4236	4468
DEBT SERVICE-										
INTEREST	12.	9569	9500	9424	9337	9239	9129	9005	8865	8707
PRINCIPAL	.70	534	603	679	766	864	974	1098	1238	1396
#TOTAL DEBT SERVICE	10.	10103	10103	10103	10103	10103	10103	10103	10103	10103
#PRE-TAX CASH FLOW	-7.	-7253	-7110	-6962	-6809	-6652	-6489	-6322	-6149	-5972
TAX CREDIT(OR LIAB) DUE TO-										
NET INCOME(LOSS)	55.	3669	3554	3421	3300	3161	3012	2853	2682	2499
DEPRECIATION	55.	1638	1638	1638	1638	1638	1638	1638	1638	1638
#TOTAL CREDITS(LIAB'S)	55.	5307	5192	5059	4939	4799	4650	4491	4320	4137
#AFTER-TAX CASH FLOW	-1.	-1946	-1918	-1893	-1871	-1852	-1839	-1831	-1829	-1835

>>>>> RETURN ON INVESTMENT ANALYSIS:

ITEM	%	YEAR 01	YEAR 02	YEAR 03	YEAR 04	YEAR 05	YEAR 06	YEAR 07	YEAR 08	YEAR 09
#GROSS SALES PRICE	10.	110000	121000	133100	146410	161051	177156	194872	214359	235795
SALE EXPENSES-										
COMMISSION	6.0	6600	7260	7986	8785	9663	10629	11692	12862	14148
ESCROW	1.0	1100	1210	1331	1464	1611	1772	1949	2144	2358
OTHER SELLING COSTS	2.0	2200	2420	2662	2928	3221	3543	3897	4287	4716
LOAN BALANCE		79466	78863	78184	77418	76554	75580	74482	73244	71849
TAXES ON GAINS	23.	721	3745	7003	10517	14312	18417	22862	27685	32916
#TOTAL SALE EXPENSES		90086	93498	97166	101111	105360	109941	114883	120219	125986
CASH RETURN-										
NET EQUITY RETURN		19914	27502	35934	45299	55691	67215	79989	94139	109809
CUM CASH FLOW (FV)	5.0	-1946	-3961	-6052	-8225	-10489	-12852	-15325	-17921	-20651
#TOTAL RETURN		17968	23541	29883	37074	45202	54363	64663	76219	89158
CASH INVESTMENT-										
DOWN PAYMENT		20500	20500	20500	20500	20500	20500	20500	20500	20500
CUM CASH FLOW (PV)	5.0	1853	3593	5228	6767	8218	9590	10891	12129	13312
#TOTAL INVESTMENT		22353	24093	25728	27267	28718	30090	31391	32629	33812
RETURN ON INVESTMENT (AFTER TAX)-										
TOTAL R-O-I (%)		-12.4	14.8	45.8	80.9	121	165	215	272	335
TOTAL I-R-R (%)		-12.4	7.16	13.4	16.0	17.1	17.7	17.8	17.8	17.7
EQUITY R-O-I (%)		-10.9	14.2	39.7	66.1	93.9	122	155	189	225
EQUITY I-R-R (%)		-10.9	6.84	11.8	13.5	14.2	14.3	14.3	14.2	14.0
RETURN ON EQUITY (%)		-12.4	28.5	27.8	20.9	18.9	17.4	16.3	15.4	14.7
CONVENTIONAL IRR (%)		-11.6	7.00	12.6	14.8	15.7	16.0	16.1	16.0	15.9

PRINTING LABELS ECONOMICALLY

By E. J. NEIBURGER

If you do a lot of label printing you may have found some disadvantages in using commercial gummed labels. One problem is that they are rather expensive, 1-3 cents each depending on quantity purchased. Another disadvantage arises because of the construction of these labels. The pre-cut labels are mounted on a waxed paper-like backing. The extra thickness may cause jamming in some printers and slipping in others. Finally, some label printing programs occasionally skip a line due to some sort of glitch. This possibility requires constant supervision of the printing process to keep the printing from sliding off the label.

There is an economical alternative to commercial labels. Use white or colored packaging tape. This is a three inch wide tape with a glue coated back side. It comes in rolls of 100 yards and costs about $5 at utility stores. Most friction feed printers can use this material. It doesn't jam readily and line skips or slippage spoil only one label and not many subsequent ones.

The only disadvantage is that the labels are single column and must be cut and moistened for use. However the conveniece in buying and storing the tape and the low cost, about 75 labels per penny, more than compensates for the time spent cutting and "licking" the labels.

DECEMBER AGENDA

Dec 12, 1981. "A" BUILDING

10:00-10:30am Opening Remarks
(Dave Alpert)
10:30-11:00am Interactive Video Disk System
(Symtec Inc)
11:00-11:30am Statement of the Month
(G. Lyle)
11:30-11:45am Break
11:45-12:00pm Nominations
(Joel Kurasch, John Hoffman)
12:00-12:30pm Sights and Sounds
(Tom Porett)
12:30-12:55pm Ask Mr. Apple
12:55-1:00pm Closing Remarks

1200 BAUD MODEMS: DON'T GET TOO EXCITED

BY: TERRY CRONIN

It's too bad that Novation's new 1200 baud modem defies superficial logic. But, unfortunately, it's not 4 times better than a 300 baud modem.

For one thing, you might not be able to find anybody to communicate with. To communicate with any 1200 baud modem, you need to hook up with a compatible modem, and here's the biggest problem of the Novation 1200 baud: you may not be able to use it. There are two types of 1200 baud modems: those designed to be compatible with Bell System 212 modems, and those that can be used with Bell 202A modems. Telenet, The Source, and Tymenet all use the 212 system. If you buy the Novation Apple-Cat II, you won't be able to access those systems at 1200 baud. They use the Bell 212 because it permits full-duplex use. The Bell 202A that Novation has used on the Apple-Cat allows only half-duplex communication. And, word is that Tymnet, for one, is getting away from ALL half-duplex systems. So the Novation couldn't even be used in the future.

So, if you're interested in hooking up to one of the big information services, you MUST choose either a 1200 baud, 212 compatible modem, or a 300 baud such as the Hayes. 1200 baud, 212 compatible modems cost between $800 and $1200-- substantially higher than even the fanciest 300 baud modems.

Furthermore, the usage or connect fees charged by the information services are much higher if you use a 1200 baud 212 compatible modem-- about 50% higher. Only if you move huge amounts of data can this cost justify itself.

And more bad news about 1200 baud modems: some services, (notably CompuServe) won't even let non-commercial users hook up with 1200 baud, only commercial accounts.

So, if you're thinking about a 1200 baud, you'd better check out all possible hook-ups. Unfortunately, if you buy a 202A compatible 1200 baud modem, you're going to have to make sure that you can talk to someone else with one. (If you're running a closed-terminal system that receives a great deal of information from one source, they MAY make sense, if you know you're restricted to talking to a similiar modem.) And if you're considering the other type of 1200 baud modem, the 212 compatible, be sure you know the possible pitfalls before you buy.

UNCLASSIFIED ADS

FOR SALE

Long Ribbon Cables for your Apple

Apple Disk Drive Data Cable
-2 feet for $10.00, $1.00/each additional foot

D.C. Hayes Micromodem II cable
5 feet for $10.00, $.50/each additional foot

Printer Extension Cable

(Epson & Others)
Custom Made $10.00+$1.00/foot
20 pin plug & 20 pin socket.

Illinois residents add 6% sales tax. Allow up to 6 weeks for delivery

CONTACT: Rob Stewart,
763 S. Wayne Place,
Wheeling, Il 60090.
Home 312-537-0544
Modem 312-537-7063

FOR SALE
Microsoft RAM card $150, Videx Keyboard Enhancer, Rev 7 keyboard or greater $90. Bob Rzadzki, 312-980-5546.

FOR SALE
Excellent condition Epson MX 80 printer with tractor feed, extra print head, interface card, get-started quantity of high quality paper and 5 disks. $525 or will trade for disk drive or (Z80 card or 80 column card) plus legitimate, assignable software--DB Master, etc. Norm Pugh 312-469-9276

COMPUTER GAMES PLAYERS
Looking for computer game players, call Bob Ritz, 742-0381

****

WANTED : For position of PRODUCT SUPPORT MANAGER located Chicagoland with knowledge Apple Basic, 6502 Assembly and Forth. Salary in 40's. Call including evenings : 317 / 255 - 6601 or send resume to : Karl Andresen, Quest Incorporated, 7401 North Edgewater Drive, Indianapolis, Indiana 46240.

****

Al Guthrie
907 Jeffery Court
Schaumburg, IL 60193
(312) 529-0455

RS-232C IS STANDARD ....ISN'T IT?

Al Guthrie

The words "RS-232C interface" have a nice, reassuring ring. The bargain hunting back-of-the-book ad reader has visions of a universally useable device, which can be plugged into any RS-232C serial interface on any computer.

Well, it aint necessarily so.

This notion of a simple, standard interface is due to a misunderstanding of just what is actually being standardized. The Electronic Industries Association (EIA) RS-232 standard is intended for the interconnection of communications devices. It has been applied, and sometimes mis-applied, to devices that had no standard of their own, such as computers and serial printers.

The standard defines the type of connector to be used, assigns the pin positions, and describes the permissable signal voltage levels. It allows wide variation in the protocol, or "handshaking", that takes place between two devices that want to talk to each other.

The connector specified is a Cinch or Cannon DB25. It is about 2 inches long and 3/8 inch wide. One side has twelve pins and the other has thirteen, so that the male and female connectors can only be connected one way. The standard says the Data Terminal Equipment (DTE) will have a cable terminated in a male connector; the Data Communications Equipment (DCE) will have a female connector. A modem, for example, is a DCE. In that case a computer connected to a modem is a DTE.

Here is where the first problem arises. If a printer is connected to an Apple, which is the DCE? It turns out the Apple thinks it is a DCE. What the printer thinks is another matter. If your printer and your serial interface board don't agree on this point, you will have to re-wire the cable or perhaps make an adapter.

The uses of all but three of the connector's pins are defined by the standard. Please note however, that just because the use of a pin is defined in the standard does not mean the pin must actually be used. In fact, only a few will ever be of concern to the Apple.

Figure 1 shows the standard pin assignments.

Pins dealing with secondary signals (12,13,14,16 and 19) are used where a sub-channel on a Communications circuit carries the control signals. Pins 8 through 10, 15, and 21 through 23 are used on switched networks. None of these are likely to be used in a printer application.

Pins 2,3 and 7 are always essential, since these carry data to and from the interconnected devices. The remaining pins allow plenty of scope for variation.

It's nice if the printer can let the Apple know when to stop sending data. Otherwise, if the printer ran out of paper, and the stream of data continued without pause, the data would wind up smudged all over the platen.

Some printers handle this problem by use of the Data Terminal Ready line (pin 20). The Apple must monitor this pin and when it is low (the voltage drops) it must stop transmission until it is high again. That sounds like a sensible solution except that not all serial printers use this scheme, and not all interface boards (the Apple Serial Interface Board among

them) pay any attention to this signal.

Other control methods are possible, such as using the Request To Send and Clear To Send lines (pins 4 and 5). The point is, that both devices must use the same scheme or nothing is accomplished.

Another method commonly employed is to send a control code over the data circuit and not use the control circuits at all. This works well providing the two devices agree on what the codes should be. Some use DC1/DC3 (Apple keyboard control Q and control S). The Apple Serial Interface prefers ETX/ACK (control C and control F).

What if you get a printer that uses a control scheme none of the available interface boards support? Or, you are stuck with a board that insists on a different protocol? As far as recognizing an out of paper or other abnormal condition, there is really nothing you can do, except be an attentive operator. The more vexing problem however, is buffer overflow.

Buffer overflow occurs because data can be transmitted much faster than it can be printed. For example, a printer with an IBM Selectric mechanism is limited to a maximum print rate of 14.9 characters per second. If you transmit at the relatively slow rate of 300 baud, characters will arrive twice as fast as they are used. Eventually the buffer, no matter how large, will be full and unable to accept more data. If the bit stream continues to flow it will spill out on the floor or somewhere, and some characters will be lost.

The only solution is to slow the transmission rate to less than the printer's rated speed and allow it to operate at less than an optimum rate.

The moral of the story is clear. Marriages may be made in heaven, but printers and interface boards are made in different factories.

Fig. 1 RS-232C PIN-OUT

Description	Pin	Pin	Description
Transmitted Data	2	1	Frame Ground
Request to Send	4	3	Received Data
Data Set Ready	6	5	Clear to Send
Received Line Signal Detector	8	7	Signal Ground
- DC Test Voltage	10	9	+ DC Test Voltage
Secondary Received Line Detector	12	11	Unassigned
Secondary Transmitted Data	14	13	Secondary Clear to Send
Secondary Received Data	16	15	Transmitter Clock
Unassigned	18	17	Received Clock
Data Terminal Ready	20	19	Secondary Request to Send
Ring Indicator	22	21	Signal Quality detect
External Transmitter Clock	24	23	Data Rate Select
		25	Unassigned

PERUSING THE PERIODICALS

by MARLYS NEWCOME

CREATIVE COMPUTING VOL 7 #11 NOV/81
* *

Author David Lubar
Title AUTHORING SYSTEM FOR THE APPLE
Page 34
Subject A review of four programs for teachers to use to write lessons

Author Helmar Herman
Title APPLESOFT COMPILERS
Page 40
Subject A comparative evaluation of four Applesoft compilers

Author Dale Archibald
Title COMPUTER AIR COMBAT
Page 58
Subject A review of a strategic air combat simulation, conclusion-a marvelous game

Author David Lubar
Title NEW GAMES FOR THE APPLE
Page 89
Subject A review of several new games

Author George Blank
Title A TOURIST'S GUIDE TO THE CYBERNETIC TOWER OF BABEL
Page 94
Subject A short discussion of ten languages available for some popular computers

Author David H. Ahl
Title PASCAL, ADA AND COMPUTER LITERACY
Page 116
Subject An interview with Art Luehrmann

Author David Lubar
Title THREE ROADS TO PASCAL
Page 124
Subject A review of three pascal books

Author Jonathan D. Ashwell
Title BUILDING A BETTER INPUT
Page 232
Subject A program that allows input with all punctuation marks

Author Donald T. Piele
Title HOW TO SOLVE IT
Page 238
Subject Another in his series of articles about teaching programming

Author Chuck Carpenter
Title APPLE CART
Page 250
Subject Getting started: hardware, software,printers

This issue also has several other articles on languages, a report on the Electronic Mail and Message Conference, and reviews of Raster Blaster, Bridge Challenger and Crypto Machine.

INTERFACE AGE VOL 6 #11 NOV/81
* *

Author Patrick and Leah O'Connor
Title GAME CORNER
Page 24
Subject A review of Robot War, an excellent game, in which you program your robot

Author Mike Dhuey
Title APPLE-ICATIONS
Page 40
Subject Some basics of Applesoft Basic

Author Larry and Jacqueline Raymond
Title CHOOSING YOUR PRINTER HARDWARE
Page 70

Author Alan J. Nayer
Title USING ABBREVIATIONS IN APPLE PASCAL
Page 98

SOFTSIDE VOL5 #1 OCT/81
* *

Author "J"
Title THE SENSUOUS PROGRAMMER
Page 17
Subject An article about working with strings in Basic

Author David Delli Quadri
Title THE YANG/YIN SYMBOL
Page 72
Subject A search for execution speed in a simple HPLOT routine

This issue includes Leyte, a simulation of the battle of Leyte Gulf and Arena of Octos, a game

COMPUTE! VOL3 #10 OCT/81
* *

Author Michael P. Antonovich
Title UNDELETABLE LINES
Page 62
Subject A method for putting lines in your program that can not be deleted unless you use the monitor

Author William R. Swinyard
Title BUDGETING ON THE APPLE
Page 72

Author Brian Nakagawa
Title SWITCHING CLEANLY FROM TEXT TO GRAPHICS
Page 82

MICRO THE 6502/6809 JOURNAL #41 OCT/81
* *

Author Sherm Ostrowsky
Title APPLESOFT MYSTERY PARAMETER
Page 12
Subject An undocumented "booby trap" in Applesoft is utilized

Author Kim G. Woodward
Title APPLE BYTE TABLE
Page 88
Subject Help for decoding byte values in Apple's memory

Author Richard C. Vile
Title APPLE BITS, PART 2
Page 94
Subject This part presents an Integer Basic program for constructing patterns

Author Dave Partyka
Title SOLAR SYSTEM SIMULATION, PART 2
Page 108
Subject This geocentric program will print information about 6 planets

NIBBLE VOL2 #6
* *

Author Glenn Teman
Title THE APPLE M. A. P. S. SYSTEM
Page 9
Subject A marketing and personal shopping system

Author Paul Irwin
Title AMPER JUMP AND TSORT
Page 21
Subject A fast machine language tag sorting program

Author Tony Dahbura
Title APPLE ARTIST
Page 43
Subject A Hi-res shapemaker

Author David E. Lieberman
Title PASCAL POINTERS AND PRINCIPLES
Page 53

This issue also contains reviews of 2 compilers, some Apple tips and utilities.

CALL-A.P.P.L.E. VOL 4 #8 OCT/81
* *

Author Dana J. Schwartz
Title HUFFIN (PASCAL TO DOS)
Page 13
Subject Converts Pascal files to DOS text files

This issue contains a special Hi-res section, 4 word processor evaluations, and a review of compilers.

BYTE VOL6 #10 OCT/81
* *

Author C. A. Johnson
Title PREPARE YOUR PROGRAM FOR PUBLICATION
Page 114

Author Wendell Brown
Title A SIMPLE IMPLEMENTATION OF MULTITASKING
Page 176
Subject This program is not written for the Apple, but the idea could be applied to the Apple

Author Gregg Williams
Title TREE SEARCHING, PART 2: HEURISTIC TECHNIQUES
Page 195

Author Dr. James Stork
Title DRAWING WITH UCSD PASCAL AND THE HIPLOT PLOTTER
Page 214

Author David Casseres
Title BITS AND BYTES IN PASCAL
Page 448
Subject Some tricks you can perform in Pascal

This issue contains a review of the IDS 460 printer and Mauro MP-250 Proac plotter plus several articles on computer networks.

PERSONAL COMPUTING VOL5 #11 NOV/81
* *

Author Martin A. Nemzow
Title ADVANCED COMPUTING
Page 94
Subject Fixed-format data entry for the Apple

APPLE ORCHARD VOL2 #3 FALL/81
* *

Author Peter C. Weiglin
Title BUILD A BETTER ERROR TRAP
Page 12

Author Joe Budge
Title PRICE WAR
Page 38
Subject A simulation to teach how market forces act on a firm

Author Val J. Golding
Tile INTEGER BASIC REGISTER LOADER
Page 53

Author David G. Huffman
Title SHAPE DEFINITION CONVERSION TABLE
Page 78
Subject A table to help you edit an already defined shape

* * * * * * * * * * * * *

ANOTHER MARQUEE

by Rik Anderson
Rt 2-Barrington Bourne
Barrington Hills, Il 60010

The thing I like about Public Domain Software is the way good ideas feed upon one another in that environment. I'm not nearly as good at coming up with original ideas as I am at taking someone else's good idea and making it better. In this case, Peter Clarke's article in the September HARVEST was the jumping-off point for my "better idea".

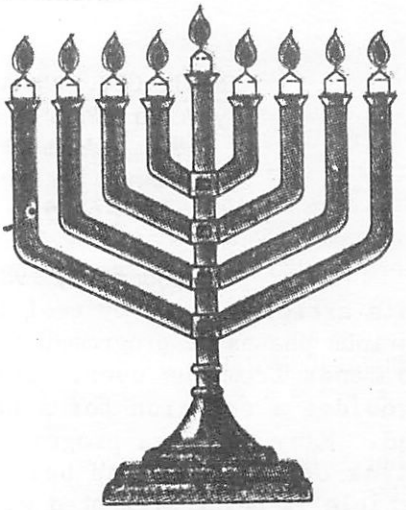
In the marquee presented by Mr. Clark the "stars" all move from right to left. On real theatre marquees the lights are flashed in such a way to give the illusion that they are racing around the sign. By adding separate strings to run along the bottom (C$) and the sides (V$), and line 1100 which causes the C$ string to run left to right, the stars appear to race around the sign. In the accompanying program there is also a fixed frame of asterisks around which the moving frame moves.

```
]LIST

1000  REM  MARQUEE
1100  HOME
1101 A$ = ".....HERE IS A MARQUEE
      ON WHICH THE 'LIGHTS' GO AR
     OUND THE SIGN....ADAPTED FRO
     M A PROGRAM BY PETER CLARKE,
      MODIFIED BY RIK ANDERSON"
1102 B$ = "     ****     ****     **
     **     ****     ****     ****
       ****     ****     ****     **
     **"
1103 Z$ = "**********************
     ****************
1104 C$ = B$
1105 V$ = "*     ***"
1107  VTAB 23: HTAB 6
1108  PRINT "(PRESS ANY KEY TO ST
     OP MARQUEE)
1110 V$ =  MID$ (V$,2) +  LEFT$ (
     V$,1):A$ =  MID$ (A$,2) +  LEFT$
     (A$,1):B$ =  MID$ (B$,2) +  LEFT$
     (B$,1)
1111 C$ =  MID$ (C$,8) +  LEFT$ (
     C$,7)
1120  VTAB 8: HTAB 1
1121  PRINT  LEFT$ (B$,40)
1122  VTAB 9: HTAB 1
1123  PRINT  MID$ (V$,1,1)Z$ MID$
     (V$,1,2)
1124  VTAB 10: HTAB 1
1125  PRINT  MID$ (V$,8,1)"*" SPC(
     36)"*" MID$ (V$,2,1)
1126  VTAB 11: HTAB 1
1127  PRINT  MID$ (V$,7,1)"* " LEFT$
     (A$,34)" *" MID$ (V$,3,1)
1128  VTAB 12: HTAB 1
1129  PRINT  MID$ (V$,6,1)"*" SPC(
     36)"*" MID$ (V$,4,1)
1130  VTAB 13: HTAB 1
1131  PRINT  MID$ (V$,5,1)Z$ MID$
     (V$,5,1)
1132  VTAB 14: HTAB 1
1133  PRINT  LEFT$ (C$,40)
1140 X =  PEEK ( - 16384): IF X >
     127 THEN  POKE  - 16368,0: GOTO
     1180
1150  GOTO 1110
1180  RETURN

]
```

SEASON'S GREETINGS



WORD LIST

1. CHRISTMAS
2. DECEMBER
3. SANTA
4. PRESENTS
5. SNOWMAN
6. MENORAH
7. EGGNOG
8. RUDOLPH
9. STOCKINGS
10. MISTLETOE
11. GELT
12. TURKEY
13. TREE
14. HANUKKAH
15. POINSETTIA
16. CHESTNUTS
17. WREATH
18. CAROLS
19. DREIDL
20. SNOW
21. HOLLY
22. REINDEER
23. CARDS
24. LATKES
25. NOEL
26. SKIING
27. FIREPLACE
28. SLEIGHRIDE
29. TOYS
30. SHOPPING
31. FRUITCAKE
32. COOKIES
33. BELLS
34. PINATA
35. DASHER
36. DANCER
37. PRANCER
38. VIXEN
39. COMET
40. CUPID
41. DONDER
42. BLITZEN
43. FAMILY
44. FRIENDS
45. WINTER

J X L F B L I T Z E N P O I T E M O C Q
H Y G T Z E M U H K S B E S W J J P N A
P E O D X B W F P A O R E M N V M C L L
S K G N I I K S S H V G N I P P O H S S
T R Y W J G H Y C B X B Z K I O D I Q T
U U C I J O O I F N L F R U I T C A K E
N T T N J T E X D M B H A K K U N A H E
T U P T I W F Z T U T Y E A Y M O E Y V
S U G E G N X Q H C T M X D L J P T V U
E N Q R I N B X R E J D U I A F R Y Q Q
H A C E I I X F A D R G V P V I E Q A N
C C L I D Q I S F I W Q A U N A C K U W
F R Y T J B D L S R W T Q C T H N L J N
E E E Y L X G S H H R Q C A B I A L S K
D C L H J X F H L G C O N I D J R C W J
A A X E S T O Z O I O I C L K R P Y X U
N L B Z O A Y L P E P Z I X F B E W D J
C P G X Q N D L D L R E B M E C E D G Q
E E Y R H K B D L S S E K T A L X P N X
R R K M B T D F Q O I G Q C V I X E N S
L I K A F D R F L L H D E Q X V C N P R
D F E G K I W W J N A M W O N S O T T D
I V R E E D N I E R W L Q H J T K C V O
E E E N S T K V S G N I K C O T S D N F
R X D Q G S B G B Z Y L K J C X F L V C
D S D X G I T P I G M K K Y B K A S I K
P B T X C X N N E T W M O U I G M D I K
S O L J R H M L E S R H H P L O I R N S
M Y E Y L P S F M S H E L O M N L A R Z
Z H G I L L L N I Y E T E I Z G Y C D R
G A N J Q O O J S S M R U N B G F N N E
E R S Z J D R K T N C V P S C E D Q K D
B O O H N U A U L X L P S E H T L N G N
V N D J K R C E E S P Y I T R M A L M O
E E D C T Q D J T N O J M T I W I I S D
K M H T A E R W O O L V T I S X K H N V
S E I K O O C N E W O Z Y A T T E B H E
R B P E Y W H Q I Q T N E M F V P S U
T H U W Y O Z J A T N A S K A E L E R N
Q E W Q O P P X X M U V W C S M E A Q E

COMPUTER GENERATED WORD PUZZLE

INPUT, GET, COMMAND?

by Robert Paarlberg
1961 Haddam Pl
Hoffman, Estates
884-6489

October 10, 1981

This article traces my evolution through the various phases of programming the Apple to get commands from the user. The final section provides a solution for a unique problem I had. Many times as programmers, we tend to think of our problems being unique. This article is being presented with the view that few problems are unique and it will hopefully provide a solution to someone else.

HISTORY

With the advent of the Apple and similar CRT (TV) based computers, programmers had to change their perspectives from designing batch programs and switch to interactive programs. Batch programs are those that typically take a transaction file and perform an activity on these batches or group of transactions. The program is started by the operator and no interaction is required. The job will perform all of its activities without any outside intervention. Much business data processing is still done on this level. The newer approach is the interactive or on-line approach. In larger installations this is used for key entry stations and performs the functions of editing and occasionally updating a master file.

Most of the programs we now produce are on-line and menu driven. A menu is simply a borrowed term from the diner where we ordered the number 5 sandwich. If we, as users of a program, select a number 5, we are telling the Apple to go to a specific routine and perform a task and then return with information and another menu. An extension of this menu technology is the use of layered menus where the user selects one option from a master menu and then selects another option from a sub menu. An example of this is where we would select the sort on the master menu. The sub menu would then offer us several sorting sequences and options. After completing the sort we would return to the sort sub menu which would allow us to continue and do another sort or go back to the master menu.

LISTING 1

The technology of the first couple of months of programming for most of us consisted of testing the input string for each valid value and branching. (see listing 1) In small applications, this is a perfectly reasonable test. It has the advantages of being straight forward, easily traceable and does not require remarks.

LISTING 2

The next step was to number the choices in the menu and use an 'ON Variable GOTO' format. This allowed the test to be made for typographical errors by setting the permissable range. (listing 2). This was an extension on the timeshare terminal comand "input (1) yes, (2) no" where a numeric input was required. I'm sure you've all seen some old converted programs that still use this structure.

LISTING 3

Further on in our development was the use of setting the value of A (for the ON Variable GOTO) by getting the ASCII value of the input prompt. (see listing 3). Line 100 takes the letter in A$ and converts it to the ASCII value-64. This means A=1, B=2 etc.

LISTING 4

Listing 4 shows an example of an input routine where the Boolean value is set. This subroutine is very handy where the program has many yes/no prompts. The subroutine in Line 30 turns A on (A=1) if the user responds with a 'Y' to the prompt.

LISTING 5

A more sophisticated approach to a menu is shown in Listing 5. This subroutine allows you to build a mnemonic menu. Line 130 gets the command and then it is compared to the command string (CM$) by the loop in Line 140. The loop will exit when A$ = position I in CM$. The value of I can then be used as the variable in Line 170.

LISTING 6

A form of screen mapping is shown in Listing 6. This program will print all of the commands. The user may then select one or more command by moving the cursor through the list. Lines 50 through 70 write the commands to the screen. Lines 100 through 220 control moving the cursor through the list. The rest of the subroutine reads the location of the cursor if 'P' was selected (Line 260), and reads memory from cursor position to cursor + 10. The result of this read is then put into a string array. At the same time the command you picked is rewritten to the screen in inverse. This lets you easily identify those

items selected. This routine is valuable when you wish to have combined commands issued. I have used this routine in a shopping list program where 80 items per aisle can be printed on the screen. The user then steps through the list picking those items he wishes to buy. When quit (Q) is selected, a report is printed, listing the items to buy by aisle.

OTHER METHODS

While there are many other variations on the first 6 listings, one more deserves mention. This type is most often used when inputting records to a file. The general approach is to first print all of the field descriptions on the screen. After this, the program brings the cursor to each field and prompts input. One of the nicer features added to this type of input is to have a series of dots or inverse blanks to show the maximum length of the field. Another is to allow exiting the routine if the escape key is pressed for any input field. For more information on this approach, look at Charles L. Hearn's article "INVERSE FIELD INPUT FOR APPLESOFT" in the June issue of CALL APPLE as well as The PROMPTER in Dakin5's Programming Aids.

This brings us to the point of this article. I was recently confronted with a program that contained over 40 legal commands. The program was designed with a command loop that tested the input string for each of the valid commands, much as listing 1. This code required almost 2K of valuble memory. The largest constraint was that the program is currently being used in over 200 installations by non-technical personnel. Changes to the program that would require training or updating the documentation had to be avoided. The solution had to be transparent to the user as well as free up memory.

COMMAND FINDER

Listing 7 is the Applesoft program that controls the Command Finder. It's object is to get a command, poke the command and the commands length into the machine language program and then execute by a call. The machine language program (Listing 8) compares the command (CMD) poked by the Applesoft program to the command string (STRING). Everytime the program hits a '/' in the string, it increments the counter (CTR).

If it matches the first letter then it compares the second position of the command to the second position of the string. This continues until one of the following conditions are met: (1) The counter equals the length of the string meaning no match was found; or (2) All characters in the command have been found in the string and the counter equals the command length. Once either of these conditions have been met, the machine language program turns control back to the Applesoft program. At this point, the value in 'CTR' ($3BB) will contain the number of the command found. If the command wasn't found 'CTR' will contain the number of all the commands in the string +1. This value can then be used in a 'On A Goto 100,200,300, etc.' format.

To increase flexibility, program has been included to allow you to input a new command string in the machine language program without having to reassemble. (Listing 10). The 'Command Poker' takes your command string and pokes it into the machine language program and then saves the new string along with the program.

APPLICATION NOTE:

The machine language program is loaded starting at the top of page 2 ($2FB). Readers should note that page 2 is the input buffer. Starting the program at this location will have no effect unless the user inputs a long command. As assembled the command would have to be over 251 characters. This is not much of a restriction during the running of the program but the user should be aware that the binary file should be loaded each time the program is executed. If this is not done, any long program code line keyed in will clobber the machine language code.

OPERATING INSTRUCTIONS

If you have an assembler, key in listing 8 replacing the command string from $340 through $3BA with your list of commands. If you do not have an assembler you will have to key in the memory dump shown in listing 9. This should then BSAVE as 'BSAVE COMMAND FINDER, A$2FB,L69'. Following this you may key in and run listing 10 which will poke your command string into memory and bsave it. The command finder program may then be run from your Applesoft program as in listing 7.

As implemented, the program is a solution to the problem of what to do when a large program allows many commands to be input. With the advent of the 16K+ expansion cards this may be

a moot question. But it is an alternative which the programmer may put in his bag of tricks.

FOOTNOTE:
Listings 1-5 were routines collected from various publications. I was unable to locate the original source or author for these. If anyone can identify an original author, I'm sure this publication will wish to acknowledge their contribution.

```
******************************
*                            *
*       LISTING # 1          *
*                            *
*                            *
******************************
```

```
10  REM   LISTING # 1
20  INPUT "COMMAND ? ";A$
30  IF A$ = "NEW" THEN  GOTO 100
40  IF A$ = "OLD" THEN  GOTO 110
50  IF A$ = "ADD" THEN  GOTO 120
60  PRINT "NOT VALID COMMAND": GOTO 20
100  PRINT "NEW": END
110  PRINT "OLD": END
120  PRINT "ADD": END
```

```
******************************
*                            *
*       LISTING # 2          *
*                            *
*                            *
******************************
```

```
10  REM    LISTING # 2
20  PRINT "1. NEW"
30  PRINT "2. OLD"
40  PRINT "3. ADD"
50  INPUT "COMMAND ";A
60  ON A GOTO 100,110,120
90  PRINT "NOT VALID COMMAND": GOTO 20
100  PRINT "NEW": END
110  PRINT "OLD": END
120  PRINT "ADD": END
```

```
******************************
*                            *
*       LISTING # 3          *
*                            *
*                            *
******************************
```

```
0  REM   LISTING # 3
0  GOTO 40
0  FOR I = 1 TO 40: PRINT "-";: NEXT I: RETURN
0  HOME : PRINT  TAB( 16)"MENU": GOSUB 30
0  VTAB 5: HTAB 15: PRINT "A. ADD"
0  HTAB 15: PRINT "B. DELETE"
0  HTAB 15: PRINT "C. PURGE"
0  HTAB 15: PRINT "D. CHANGE"
0  VTAB 15: PRINT "TOUCH KEY OF YOUR CHOICE->";
0C  GET A$: PRINT A$:A1 =  ASC (A$) - 64: IF A1 < 1 OR
    A1 > 4 THEN  GOTO 40
10  ON A1 GOTO 120,130,140,150
20  PRINT "ADD": END
30  PRINT "DELETE": END
40  PRINT "PURGE": END
50  PRINT "CHANGE": END
```

```
******************************
*                            *
*       LISTING # 4          *
*                            *
*                            *
******************************
```

```
10  REM   LISTING # 4
20  GOTO 100
30 A = 0: GET A$: PRINT A$: IF  LEFT$ (A$,1) = "Y" THEN A = 1
   : RETURN
40  RETURN
100  PRINT "DO YOU WANT TO CONTINUE";: GOSUB 30: ON A GOTO 120
110  PRINT : GOTO 100
120  PRINT "IS ALL THE DATA CORRECT";: GOSUB 30: ON A GOTO 140
130  PRINT : GOTO 120
140  PRINT "CONTINUE"
```

```
******************************
*                            *
*       LISTING # 5          *
*                            *
*                            *
******************************
```

```
10  REM    LISTING # 5
20 CM$ = "BIADCE"
30  HOME
40  VTAB 2: PRINT  TAB( 15)"MASTER MENU"
45  POKE 32,5
50  VTAB 5: PRINT "B - BEGIN PROCESSING"
60  PRINT "I - INITIALIZE NEW FILE": PRINT
70  PRINT "A - ADD A RECORD"
80  PRINT "D - DELETE A RECORD"
90  PRINT "C - CHANGE A  RECORD"
100  PRINT "E - END PROCESSING"
120  VTAB 20: PRINT "PRESS LETTER FOR CHOICE  ";:
130  GET A$: PRINT A$
140  FOR I = 1 TO 6: IF  LEFT$ (A$,1) =  MID$ (CM$,I,1)
     THEN 170
150  NEXT I
160  GOTO 110
170  POKE 32,0: ON I GOTO 200,300,400,500,600,700
200  PRINT "BEGIN": END
300  PRINT "INITIALIZE": END
400  PRINT "ADD": END
500  PRINT "DELETE": END
600  PRINT "CHANGE": END
700  PRINT "END": END
```

WWW

```
*****************************
*                           *
*        LISTING # 6        *
*                           *
*                           *
*****************************
```

```
10  REM   LISTING # 6
20  HOME : INVERSE : POKE 37,2: POKE 36,13: PRINT "MENU": DIM CC$(20)
30  NORMAL
40  POKE 36,10: VTAB 4
50  FOR J = 1 TO 15
60  POKE 36,10: PRINT IX:IX = IX + 1
70  NEXT J
80 : VTAB 23: PRINT "P=PICK: B=GO BACK: Q=QUIT  S=SKIP": GOTO 100
90 : POKE 32,11: POKE 33,20: POKE 37,22
100  REM  SELECT ROUTINE
110  CALL  - 998
120  IF  PEEK (37) < > 1 THEN  GOTO 110
130  CALL  - 998
140  FOR J9 = 3 TO 17
150  VTAB J9 + 1
160  HTAB 11
170  GET B9$
180  IF B9$ =  CHR$ (83) THEN  GOTO 220
190  IF B9$ =  CHR$ (80) THEN  GOSUB 260: REM    NUMBER P -PICK
200  IF B9$ =  CHR$ (66) THEN  GOTO 100
210  IF B9$ =  CHR$ (81) THEN  GOTO 360: REM   NUMBER Q - PRINT
220  NEXT J9
230  GOTO 100
240  GET B9$: IF B9$ =  CHR$ (66) THEN  GOTO 100
250  GOTO 360
260 L1 =  PEEK (40):L2 =  PEEK (41) * 256:
270 C = C + 1
280 L3 = L1 + L2
290  FOR I6 = L3 + 10 TO L3 + 20
300 CC$(C) = CC$(C) +  CHR$ ( PEEK (I6))
310 PN =  PEEK (I6)
320  IF PN > 63 THEN PN = PN - 64: GOTO 320
330  POKE I6,PN
340  NEXT I6
350  POP : GOTO 220
360  END
```

SPECIAL INTEREST GROUP NEWS

SIG INFORMATION CONTACT LIST:

Business———— Eric Stral 312-885-1941

Education----------Phyllis Geren 312-945-2624, Ann Baldridge 312-529-2023

Stock, Options, Commodities——Harry Maram 312-764-1847, Bob Werner 312-433-0478

Pascal————Ken Nestle 312-620-7745

Modem————Terry Cronin 312-289-6392

SIG COMING EVENTS

Education SIG- no December meeting.

Stock, Options, Commodities SIG will hold their next meeting Dec 17, 7:30 pm at MGCC building, 8944 N. Austin, Morton Grove. For more information call Bob Werner 312-433-0478.

Business SIG-no December meeting.

```
*****************************
*                           *
*        LISTING # 7        *
*                           *
*                           *
*****************************
```

```
10  REM    COMMAND FINDER A/S
15  REM    CALL 763
20 D$ =  CHR$ (4)
30  PRINT D$"BLOAD COMMAND FINDER.O,A$2FB"
40  POKE 957,123: REM      STRLEN
50  INPUT "COMMAND ?";A$
55  REM   POKE VALUES TO TABLE
60  POKE 975, LEN (A$): REM  COMMAND LENGTH 'CMDLEN'
70  FOR I = 1 TO  LEN (A$): POKE 959 + I, ASC ( MID$ (A$,I,1)): NEXT
80  CALL 763
90 B =  PEEK (955)
95  REM  THE NUMBER OF GOTO LINES IN 100 MUST EQUAL THE NUMBER OF COMMANDS IN THE STRING

100  ON B GOTO 200,300,400,500,600,700,800,900
110  FLASH : PRINT A$;: NORMAL : PRINT " -- NOT FOUND";
120  GOTO 50
200  GOSUB 1000: GOTO 50
300  GOSUB 1000: GOTO 50
400  GOSUB 1000: GOTO 50
500  GOSUB 1000: GOTO 50
600  GOSUB 1000: GOTO 50
700  GOSUB 1000: GOTO 50
800  GOSUB 1000: GOTO 50
900  GOSUB 1000: GOTO 50
1000  INVERSE :A1 =  PEEK (958) -  LEN (A$):AA$ = ""
1010  FOR I = 1 TO  LEN (A$):A2 =  PEEK (831 + A1 + I):A2$ =  CHR$ (A2):AA$ = AA$ + A2$:
      NEXT
1020  PRINT AA$: NORMAL : RETURN
```

```
                    ********************************
                2   *                              *
                3   *      COMMAND FINDER          *
                4   *                              *
                5   * FINDS LOCATION OF COMMAND     *
                6   * 'CMD' OF 'CMDLEN' IN STRING   *
                7   * 'STRING' OF VALID COMMANDS.   *
                8   * RETURNS WITH (ON.N.GOTO)      *
                9   * FORMAT NUMBER IN 'CTR'        *
                10  * IF NOT FOUND 'CTR' WILL       *
                11  * CONTAIN # OF COMMANDS IN      *
                12  * STRING +1.                    *
                13  *                              *
                    ********************************
                15  STRING   EQU  $340   ; 1ST LOCATION OF STRING
                16  CMD      EQU  $3C0   ; 1ST LOCATION OF COMMAND
                17  CMDPTR   EQU  $3BF   ; POINTER TO POSITION WITHIN COMMAND
                18  STRPTR   EQU  $3BE   ;  POINTER TO POSITION WITHIN STRING
                19  STRLEN   EQU  $3BD   ;LENGTH OF STRING
                20  CMDLEN   EQU  $3CF   ; LENGTH OF COMMAND
                21  CTR      EQU  $3BB   ;LINE NUMBER OFFSET COUNTER
                22           ORG  $2FB        ;  LOCATION TO ASSEMBLE AT
                    ********************************
                24  *                              *
                25  *     ---- NOTE ----           *
                26  * COMMAND  (STRING TO SEARCH    *
                27  * FOR) SHOULD BE POKED IN LOC   *
                28  * $3C0 AND ITS LENGTH IN $3CF   *
                29  *                              *
                30  * THE COUNTER WHICH WILL GIVE   *
                31  * THE 'ON GOSUB' VALUE WILL BE  *
                32  * IN 'CTR'  ($3BB).             *
                33  *                              *
                    ********************************
02FB: A2 00     35           LDX  #0      ;RESET SEARCH START POINTER
02FD: 8E BB 03  36           STX  CTR     ;RESET COUNTER
0300: A9 2F     37  NXTPOS   LDA  #$2F
0302: DD 40 03  38           CMP  STRING,X ;DOES STRING CHAR = /
0305: F0 33     39           BEQ  CNT     ;YES, GOTO INCREMENT CTR
0307: AD C0 03  40  RTN      LDA  CMD     ;LOAD FIRST CHAR OF COMMAND
030A: DD 40 03  41           CMP  STRING,X   ;=CURRENT STRING ELEMENT
030D: F0 07     42           BEQ  CHECK       ; IF YES, CHECK FOR REST OF MATCH
030F: E8        43  NXTSTR   INX        ; INCREMENT SEARCH START COUNTER
0310: EC BD 03  44           CPX  STRLEN      ; IS IT EQUAL TO STRING LENGTH
0313: D0 EB     45           BNE  NXTPOS      ; NO, CHECK NEXT STRING POSITION
0315: 60        46           RTS         ; RETURN, ALL CHR$ CHECKED
0316: 8E BE 03  47  CHECK    STX  STRPTR ; LET TEMPORARY POINTER = STRING POINTER
0319: A9 00     48           LDA  #0
031B: 8D BF 03  49           STA  CMDPTR      ;RESET COMMAND POINTER
031E: EE BE 03  50  CHKLP    INC  STRPTR ; INCREMENT TEMPORARY POINTER
0321: EE BF 03  51           INC  CMDPTR      ; INCREMENT COMMAND POINTER
0324: AC BF 03  52           LDY  CMDPTR
0327: CC CF 03  53           CPY  CMDLEN      ; DOES COMMAND POINTER = COMMAND LENGTH
032A: F0 0D     54           BEQ  FOUND       ; IF YES, COMMAND MATCHED
032C: B9 C0 03  55           LDA  CMD,Y    ; LOAD COMMAND ELEMENT.
032F: AC BE 03  56           LDY  STRPTR
0332: D9 40 03  57           CMP  STRING,Y    ; COMPARE TO STRING CHR$
0335: D0 D8     58           BNE  NXTSTR      ; IF NO MATCH, CHECK NEXT STRING CHR$
0337: F0 E5     59           BEQ  CHKLP       ; IF MATCH, CHECK NEXT CHR$
0339: 60        60  FOUND    RTS         ; DONE. RETURN
033A: EE BB 03  61  CNT      INC  CTR      ; CTR=CTR+1
033D: 4C 07 03  62           JMP  RTN         ; GO BACK
```

```
SYMBOL TABLE:

CHECK   = $0316  42 42 45 47 58 59
CHKLP   = $031E  50 59
CMD     = $03C0  6 16 40 55
CMDPTR  = $03BF  17 49 51 52
CMDLEN  = $03CF  6 20 53
CNT     = $033A  39 61
CTR     = $03BB  9 10 21 32 36 39 61
                 61
FOUND   = $0339  10 54 60
NXTPOS  = $0300  37 45
NXTSTR  = $030F  43 58
RTN     = $0307  40 62
STRING  = $0340  6 7 12 15 15 18 19
                 26 38 38 41 41 44 45
                 47 57 57 58
STRPTR  = $03BE  18 47 50 56
STRLEN  = $03BD  19 44
```

```
*2FB.3BA

02FB- A2 00 BE 00 03
0300- A9 2F DD 40 03 F0 33 A9
0308- D0 03 BD 40 03 F0 07 E6
0310- EE BD 03 0C E9 60 BE 4E
0318- 03 A9 00 8D BF 03 EE BE
0320- 03 EE BF 03 AC BF 03 CC
0328- CF 03 F0 0B B9 D0 03 AC
0330- BE 03 D9 40 03 D0 D8 F0
0338- E5 60 EE B8 03 4C 07 03
0340- 2A 2A 2A 2A 2A 2A 2A 2A
0348- 2A 2A 2A 2A 2A 2A 2A 2A
0350- 2A 2A 2A 2A 2A 2A 2A 2A
0358- 2A 2A 2A 2A 2A 2A 2A 2A
0360- 2A 2A 2A 2A 2A 2A 2A 2A
0368- 2A 2A 2A 2A 2A 2A 2A 2A
0370- 2A 2A 2A 2A 2A 2A 2A 2A
0378- 2A 2A 2A 2A 2A 2A 2A 2A
0380- 2A 2A 2A 2A 2A 2A 2A 2A
0388- 2A 2A 2A 2A 2A 2A 2A 2A
0390- 2A 2A 2A 2A 2A 2A 2A 2A
0398- 2A 2A 2A 2A 2A 2A 2A 2A
03A0- 2A 2A 2A 2A 2A 2A 2A 2A
03A8- 2A 2A 2A 2A 2A 2A 2A 2A
03B0- 2A 2A 2A 2A 2A 2A 2A 2A
03B8- 2A 2A 2A
```

```
                        65  ********************************
                        66  *                              *
                        67  *       TABLE OF COMMANDS      *
                            *                              *
                            ********************************
0340: 2F 45 4E 69           ASC  '/END/'
0345: 4E 45 57 70           ASC  'NEW/'
0349: 50 52 49 71           ASC  'PRINT/'
034F: 4D 41 54 72           ASC  'MATRIX/'
0356: 4F 4C 44 73           ASC  'OLD/'
035A: 52 45 50 74           ASC  'REPLACE/'
0362: 53 41 56 75           ASC  'SAVE/'
0367: 53 4F 4C 76           ASC  'SOLVE/'
036D: 4C 49 53 77           ASC  'LIST/'
0372: 49 4E 56 78           ASC  'INVOICE/'
037A: 43 4F 53 79           ASC  'COST/'
037F: 50 52 49 80           ASC  'PRINTER ON/'
038A: 50 52 49 81           ASC  'PRINTER OFF/'
0396: 50 41 47 82           ASC  'PAGE/'
039B: 4D 41 52 83           ASC  'MARK-UP/'
03A3: 41 44 44 84           ASC  'ADD/'
03A7: 44 45 4C 85           ASC  'DELETE/'
03AE: 43 48 41 86           ASC  'CHANGE/'
03B5: 50 55 52 87           ASC  'PURGE/'
                        88  SYM

--- END ASSEMBLY ---

TOTAL ERRORS: 0

192 BYTES GENERATED THIS ASSEMBLY
```

```
**********************************
*                                *
*        LISTING # 10            *
*                                *
*                                *
**********************************

10  REM   COMMAND STRING POKE(R)

15 D$ =  CHR$ (13) +  CHR$ (4): REM  CR+CNTL D
20  HOME : PRINT  TAB( 12)"COMMAND STRING"
30  GOSUB 600
110  VTAB 5: FOR J = 1 TO 40: GOSUB 900: ON E GOTO 150: NEXT
120  VTAB 9: FOR J = 1 TO 40: GOSUB 900: ON E GOTO 150:: NEXT
130  VTAB 15: FOR J = 1 TO 40: GOSUB 900: ON E GOTO 150: NEXT
135  VTAB 20: FOR J = 1 TO 3: GOSUB 900: ON E GOTO 150: NEXT
150  HOME : GOSUB 600: VTAB 5: PRINT  LEFT$ (ST$,40): VTAB 9: PRINT  MID$ (ST$,41,40): VTAB
     15: PRINT  MID$ (ST$,80,40): VTAB 20: PRINT  MID$ (ST$,121,3): PRINT "LENGTH = "; LEN
     (ST$)
155  IF  LEN (ST$) > 123 THEN  FLASH : PRINT "MAX LENGTH IS 123": NORMAL
160  PRINT "DO YOU WISH TO CHANGE ANYTHING ? ";: GOSUB 920: ON A GOTO 200
170  GOTO 500
200  HOME : VTAB 5: PRINT ST$: VTAB 10: PRINT "USE 'IJKM' AND ARROW KEYS TO EDIT STRING"
```

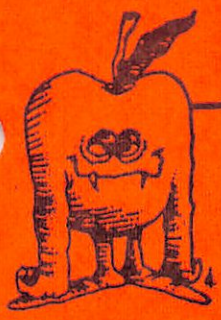
```
210  PRINT "PRESS RETURN TO ACCEPT STRING"
220  VTAB 5: INPUT "";ST$
230  GOTO 150
500  REM  POKE IN MEMORY
510  HOME : PRINT  TAB( 15)"BINARY ROUTINE"
520  PRINT : PRINT : PRINT "YOU MUST NOW PROVIDE THE NAME OF THE    BLOAD FILE YOU WISH
     TO SAVE 'COMMAND    FINDER' AND YOUR COMMAND STRING UNDER"
530  PRINT : PRINT "FOR EXAMPLE, IF YOU ARE SAVING A STRING FOR A PROGRAM CALLED 'UTILIT
     IES', INPUT THIS TITLE AND THE PROGRAM WILL SAVE IT AS A BINARY FILE UNDER": PRINT
     : PRINT "       'UTILITIES.CMDSTRING'"
540  VTAB 20: INPUT "PROGRAM NAME ? ";P$
585  PRINT D$"BLOAD COMMAND FINDER,O,A$2FD"
586  FOR I = 832 TO 954
590  POKE I, ASC ( MID$ (ST$,I - 831,1)): NEXT
595  POKE 957,123: REM   LENGTH OF STRING
598  PRINT D$"BSAVE";P$;",CMDSTRING,A$2FB,L$0C0"
599  END
600  REM  STRING LENGTH MASK
605  INVERSE : VTAB 3: FOR I = 10 TO 40 STEP 10: PRINT  TAB( I)I / 10;: NEXT : POKE 36,0

610  VTAB 4: FOR I = 1 TO 4: PRINT "1234567890";: NEXT
620  VTAB 7: FOR I1 = 50 TO 80 STEP 10: PRINT  TAB( I1 - 80)I1 / 10;: NEXT : POKE 36,0
630  VTAB 8: FOR I1 = 1 TO 4: PRINT "1234567890";: NEXT
640  VTAB 13: FOR I1 = 90 TO 120 STEP 10: PRINT  TAB( I1 - 80)I1 / 10;: NEXT : POKE 36,0

650  VTAB 14: FOR I1 = 1 TO 4: PRINT "1234567890";: NEXT
660  VTAB 18: PRINT "12"
670  VTAB 19: PRINT "123": NORMAL : RETURN
899  END
900  GET A$: PRINT A$;:C$ = A$:ST$ = ST$ + C$: IF A$ =  CHR$ (27) THEN E = 1: RETURN
910  RETURN
920 A = 0: GET A$: PRINT A$: IF  LEFT$ (A$,1) = "Y" THEN A = 1: RETURN
930  RETURN
```

DECEMBER

SUN	MON	TUES	WED	THUR	FRI	SAT
FQ 4	FM 11	1	2	3	4	5
6	7	8	9	10	11	12 NIAUG GENERAL MEETING
13	14 DEADLINE FOR ARTICLES IN HARVEST	15 OFFICERS MEETING	16	17 STOCK,OPTIONS, COMMODITIES,SIG	18	19
20	21 Hanukah	22	23	24	25 Christmas Day	26
27	28	29	30	31	LQ 18	NM 26



northern illinois apple users group

N.I.A.U.G. 2ND ANNUAL PROGRAM CONTEST
PROGRAM ENTRY FORM

FORM E11-7-81

DATE _____

NAME _____

ADDRESS _____

CITY _____

STATE _____ ZIP _____

PROGRAM NAME _____

RESUBMISSION ON NIAUG DISK NO. _____ DATE _____

FILES TO INCLUDE WITH PROGRAM IF ANY _____

PROGRAM CLASSIFICATION _____

BREIF PROGRAM DESCRIPTION _____

THE PROGRAM SUBMITTED HEREIN IS MY OWN ORIGINAL WORK AND IS BEING DONATED TO THE NORTHERN ILLINOIS APPLE USERS GROUP. PERMISSION IS GRANTED TO THE NORTHERN ILLINOIS APPLE USERS GROUP TO DISTRIBUTE THE ABOVE DESCRIBED PROGRAM IN ACCORDANCE WITH ITS CONSTIIUTION AND BYLAWS.

SIGNED _____

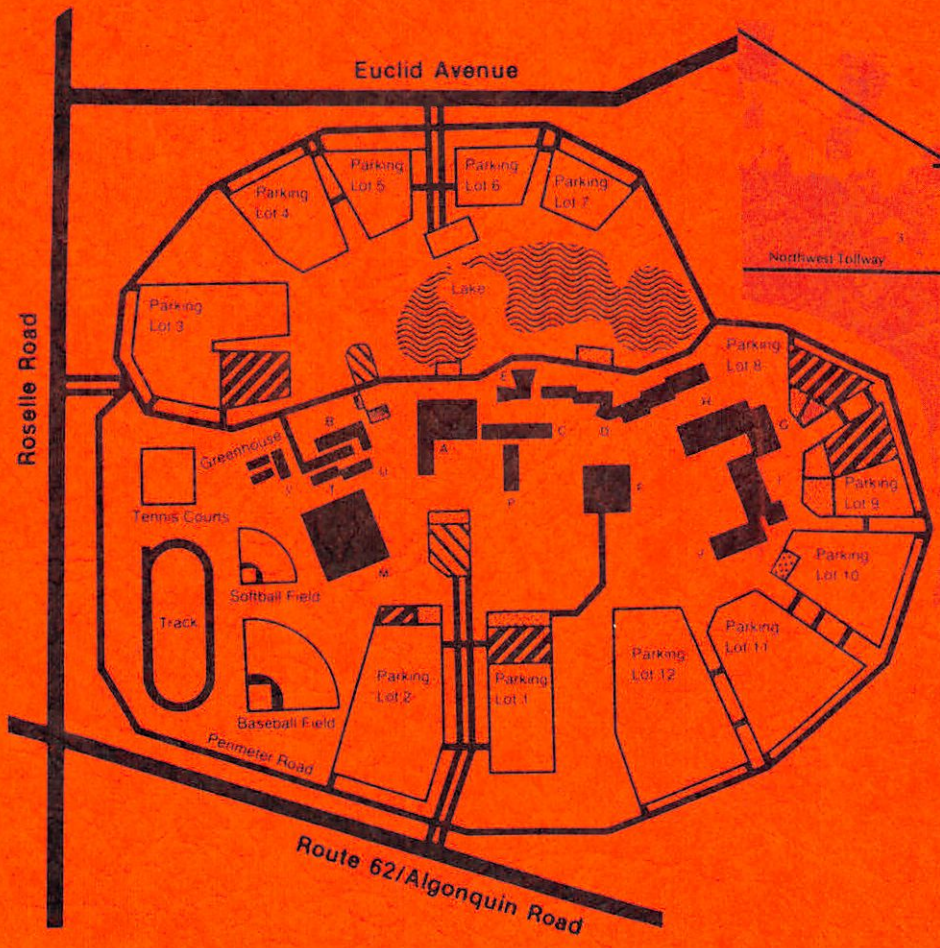
DATE _____

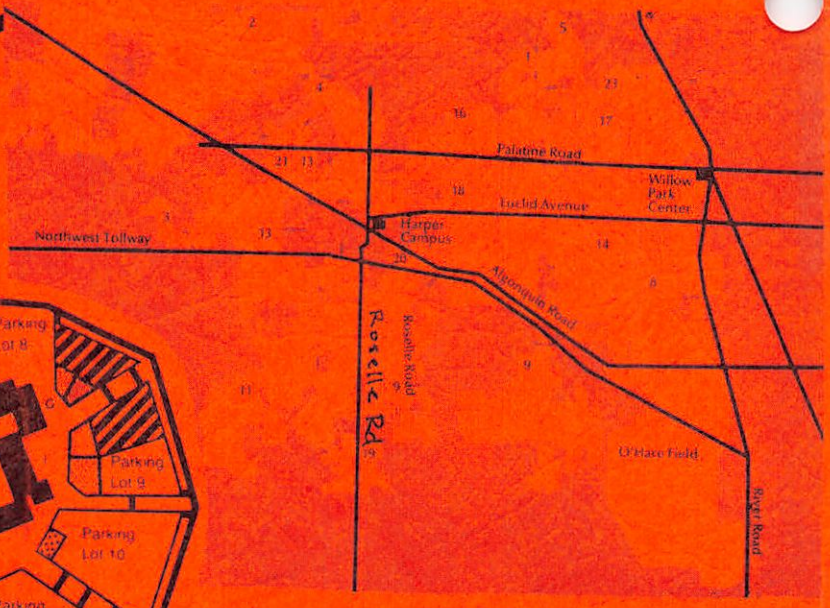
THE HARVEST NEWSLETTER
N.I.A.U.G.
1015 S. RIDGE AVE,
ARLINGTON HEIGHTS, IL.
60005.

BULK MAIL
US POSTAGE PAID
PERMIT 170
MT PROSPECT IL

MICHAEL L. ROBINS 265
1300 S. ELMHURST RD. #311
MOUNT PROSPECT, IL 60056

POSTMASTER - CONTENTS DATED
- RETURN POSTAGE GUARANTEED
- FORWARDING REQUESTED
- ADDRESS CORRECTION REQUESTED





MEETINGS

Meetings are regularly held on the the first Saturday of each month unless otherwise noted in room E106 Harper College, Palatine, Il starting at 10:00am.